Algorithm Performance Comparison

Optimal Plans

***Air Cargo Problem 1**:*

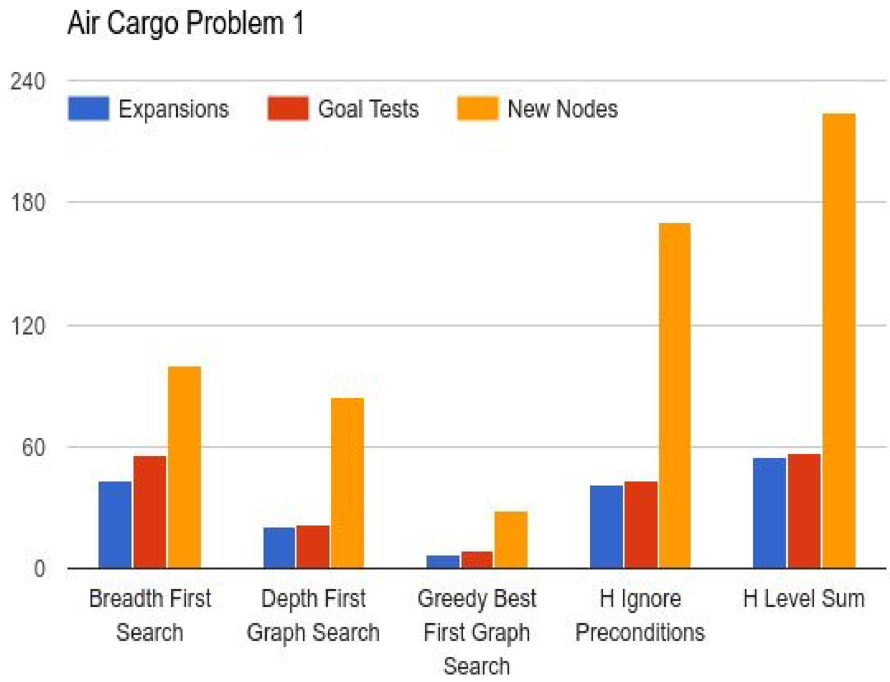
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Fly(P1, SFO, JFK)
Fly(P2, JFK, SFO)
Unload(C1, P1, JFK)
Unload(C2, P2, SFO)

***Air Cargo Problem 2**:*

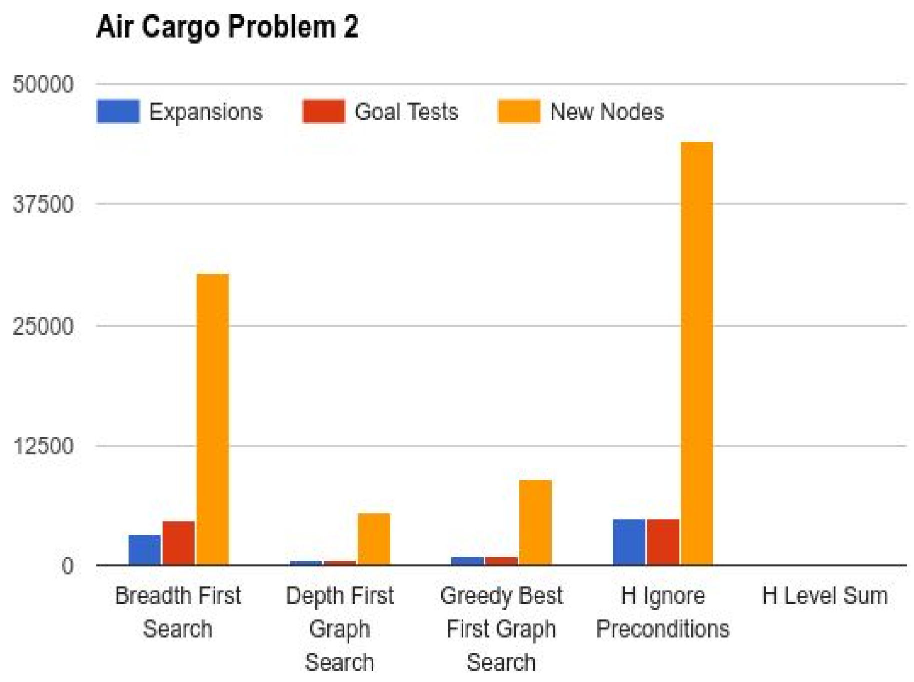
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Load(C3, P3, ATL)
Fly(P2, JFK, SFO)
Unload(C2, P2, SFO)
Fly(P1, SFO, JFK)
Unload(C1, P1, JFK)
Fly(P3, ATL, SFO)
Unload(C3, P3, SFO)

***Air Cargo Problem 3**:*
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Fly(P2, JFK, ORD)
Load(C4, P2, ORD)
Fly(P1, SFO, ATL)
Load(C3, P1, ATL)
Fly(P1, ATL, JFK)
Unload(C1, P1, JFK)
Unload(C3, P1, JFK)
Fly(P2, ORD, SFO)
Unload(C2, P2, SFO)
Unload(C4, P2, SFO)

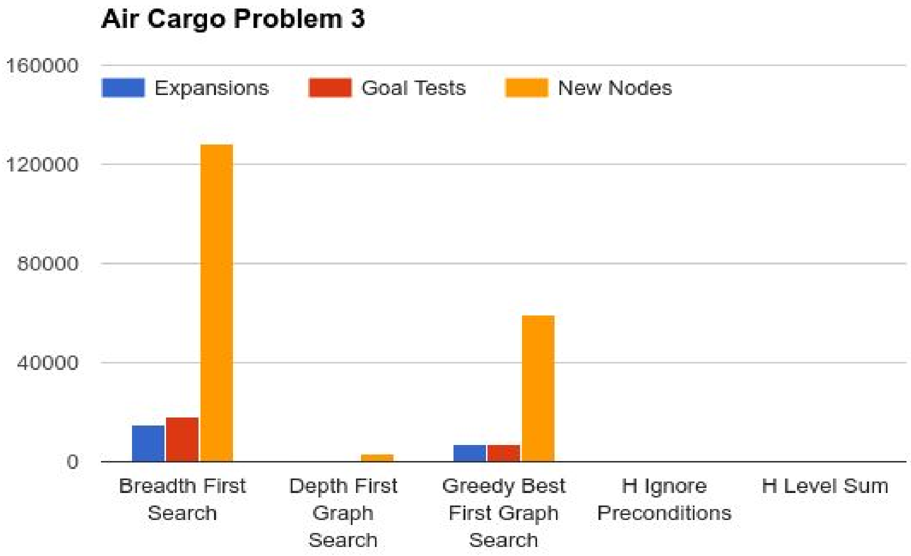
Air Cargo Problem 1	Expansions	Goal Tests	New Nodes	Time
Breadth First Search	43	56	100	0.048270189
Depth First Graph Search	21	22	84	0.023087568
Greedy Best First Graph Search	7	9	28	0.007047253996
H Ignore Preconditions	41	43	170	0.06574785701
H Level Sum	55	57	224	4.644614502



Air Cargo Problem 2	Expansions	Goal Tests	New Nodes	Time
Breadth First Search	3343	4609	30509	25.30894442
Depth First Graph Search	624	625	5602	6.293695109
Greedy Best First Graph Search	998	1000	8982	13.04724392
H Ignore Preconditions	4853	4855	44041	123.363404
H Level Sum	N/A	N/A	N/A	N/A



Air Cargo Problem 3	Expansions	Goal Tests	New Nodes	Time
Breadth First Search	14663	18098	128554	163.1873488
Depth First Graph Search	408	409	3364	2.533410296
Greedy Best First Graph Search	6943	6954	59435	166.7558041
H Ignore Preconditions	N/A	N/A	N/A	N/A
H Level Sum	N/A	N/A	N/A	N/A



Analysis

All above data were produced by using 3 different non-heuristic searches and 2 heuristic search with a limited run time of 10 mins.

According to the follow data, non-heuristic search ran faster when using DFS in general as the number of node expansion is lower than the number of nodes expanded from using other algorithms. Although it can find the right path in shorter time, it does not necessarily find the best path. It is possible that DFS can go on an unfavorable path which will end up having an extremely long path. BFS does use more time to explore and expand more nodes but it ends up with the best path among all 5 search algorithms. Greedy Best First Search is in between DFS and BFS in term of a number of nodes expanded, test, created and time elapsed.

In terms of heuristic searches, both algorithms are able to complete in the problem 1. Only A* with Ignore Preconditions is able to finish the search in 10 minutes. By comparing to the result from both algorithms performed on problem 1, we observe that A* with Ignore Preconditions has less number of node expanded and the number of goals tested compare to A* with Levelsum. It also has relatively lower time elapsed. According to the result, A* with Ignore Preconditions is a better heuristic search algorithm to to use. The likely reason A* with Levelsum have a higher execution time is because it needs to generate a planning graph which increases the complexity of the algorithm.

Conclusion:

After comparing the execution time and number of node expanded, tested and created, BFS has been identified as the better algorithm among all that we tested. In terms of heuristic functions, it appeals A* with Ignore Precondition is a better choice. It is able to find a path of equal length as the one found by A* with Levelsum but with much less time elapsed.